University of Technology Darmstadt, Germany
Department of Computer Science
IT Security Group

Seminar: Sicherheit in drahtlosen Sensornetzen
WS 2005/06

Topic:

Detecting Misbehavior in Wireless Sensor Networks

(December 20, 2005)

Supervisor:

Dipl.Inf. Christoph Krauß

Autor: Nils Knappmeier
Studiengang: Informatik

Contents

1 Introduction **3**

1.1 Wireless Sensor Networks 3

1.2 Security 4

1.3 The need for misbehavior detection 4

2 Misbehavior detection, reaction and tolerance **5**

2.1 Attack opportunities 5

2.2 Detecting misbehavior 7

2.3 Reacting on misbehavior 7

3 An intrusion detection system for wireless sensor networks **8**

3.1 Ordinary intrusion detection 8

3.2 Architecture 8

3.3 Detection algorithm 9

3.4 Simulation setup 10

3.5 Simulation results 11

3.5.1 The size of the message buffer 11

3.5.2 Reliably detectable attacks 11

3.5.3 Falsely identified attacks 12

3.5.4 Energy consumption 12

4 Statistical en-route filtering of injected false data **13**

4.1 The key distribution scheme 13

4.2 Report generation 14

4.3 En-route filtering 14

4.4 Evaluation 15

5 Assessment and further ideas **16**

5.1 Improvements to the IDS 16

5.2 Remarks on the en-route filtering 17

5.2.1 Key distribution 17

5.2.2 Countering blackhole and selective forwarding 18
5.3 Combining en-route filtering with intrusion detection 18
5.4 Unaddressed problems 19

Chapter 1

Introduction

This chapter gives a brief introduction to the topic of misbehavior in wireless sensor networks and describes the structure of this work. Section 1.1 gives a very brief introduction to wireless sensor networks and their characteristics, section 1.2 briefly displays the common security problems in sensor networks and section 1.3 outlines the reasons to deal with misbehavior detection.

Chapter 2 then gives an broad overview over several detection and reaction mechanisms.

Chapter 3 presents an intrusion detection system for wireless sensor networks. It is worth being described in more detail, because it deals with a broad range of attack forms, while other works primarily concentrate on a single attack.

Chapter 4 presents a system to filter false information. This work is important, because it expects a form of attack that is not mentioned in chapter 3. It also presents a way of dealing with intrusion automatically rather than only detecting it.

Finally, my own remarks and ideas will be presented in chapter 5

1.1 Wireless Sensor Networks

Wireless sensor networks usually consist of a number of sensor nodes being deployed in a potentially hostile environment. These nodes do not have a lot of computing power and run on battery, thus their energy resources are highly constraint. Each node has two functions

1. Perform measurements using the integrated sensors and send them towards a sink, which gathers the measurements of the whole network.

2. Forward measurements of other sensors towards the sink.

In order to conserve energy, the third task is to aggregate measurements from different sensors and forward the resulting measurement to the sink. The main

problems of sensor networks are the lack of computing and storage resources and the limited energy.

1.2 Security

These constraints have an impact of the security functions of a sensor network in the sense that not all crypocraphic methods are applicable in this area. There are a number of problems such as key-distribution, lack of public key cryptography and an attacker potentially having more computer power and energy than the sensors. This is said only to present a brief picture of the area of security in wireless sensor networks.

1.3 The need for misbehavior detection

Even if the commmunication between sensor nodes is encrypted and authenticated, there is the danger of an attacker compromising a node physically. Sensore are potentially positioned in a hostile area, so an attacker may have physical access to the nodes. If the attacker succeeds in extracting keys and other security-relevant information, he can authenticate himself as a node and perform the same kinds of attacks that are possible on an unsecured sensor network.

Additionally, a sensor node may simply misfunction. Such networks failure should also be dealt with.

This does not render cryptography useless. As an example, the intrusion detection system presented in chapter 3 assumes that all sensor nodes are uniquely identifiably. This can only be achieved by some kind of cryptographic authentication. However, there is also a need for additional mechanisms to identify misbehaving nodes and ensure the network's functionality if a node has been compromised.

Chapter 2

Misbehavior detection, reaction and tolerance

This chapter gives an overview over several works on the topic of misbehavior detection. Section 2.1 summarizes the attacks mentioned in these works, their characterics and relation to conventional network failures. Section 2.2 briefly describes methods of data aquisition and analysis in order to identify potential attacks.

Different works in the area of misbehavior detection in sensor networks have different focuses on the topic and different approaches to deal with misbehavior. Although it is never explicitly stated, it is often implied, that the detection of misbehavior is not enough, but that it is also important to handle misbehaving nodes, once they are detected. The reason is, that sensor nodes have a higher level of autonomy than traditional networks and even mobile ad-hoc networks. Once deployed, there is no user sitting next to the node, who could be warned of an intrusion taking place. The warning has to be routed over the same medium as the conventional data and is subject to the same attacks. Therefore section 2.3 gives an overview over different ways of dealing with misbehavior.

2.1 Attack opportunities

As said in section 1.3 we assume that the attacker has the opportunity to run attacks using stolen cryptographical data from compromised nodes. The following attacks are mentioned in the referenced works:

- **Message delay:** A node does not forward a message immediately but only after a certain delay time.
- **Wormhole:** An attacker creates a faster connection through the network using more radio power. It then controls the connection that most other

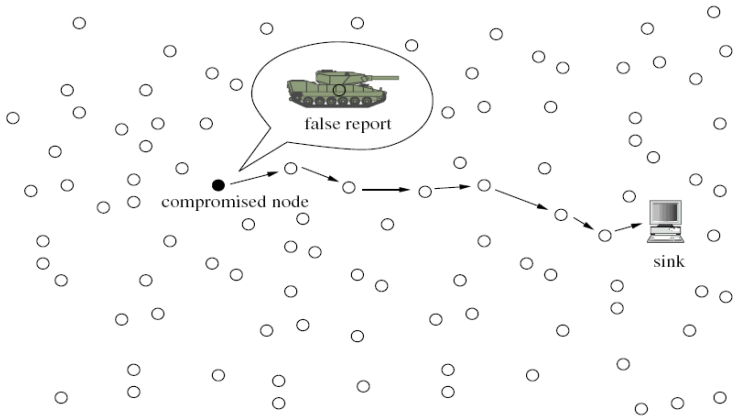

Figure 2.1: Example for the injection of false data into a sensor network ([1])

nodes will use and can run other kinds of attacks on packets passing through.

- **Message repetition:** An attacker repeatedly forwards the same message
- **Jamming:** An attacker floods the network with packets and causes collisions in order to make communication between nodes impossible. This can be mistaken with normal message collisions as they happen occasionally in sensor networks.
- **Data alteration:** An attacker forwards modified data packets. This also can occasionally happen as a network failure.
- **Blackhole and selective forwarding:** An attacker claims to be able to forward packets, but then simply drops them completely or partly. This attack can be mistaken for the occasional message loss that happens in a sensor network.
- **Exhaustion attack:** An attacker continuously sends messages in order to raise the power consumption of the forwarding nodes.
- **Injection of false data:** An attacker injects fabricated measurements into the network which is then routed to the sink.

The injection of false data is mentioned in [1]. The other attacks are taken from [2].

2.2 Detecting misbehavior

The key approach to misbehavior detection is redundancy. False data injection can be detected by comparing measurements with sensor reading from nearby sensor nodes (see chapter 4). The data forwarding behavior can be verified by nearby nodes listening in promiscous mode. This either happens on selected node (as in [2]) or the every single node in the network.

A method that is only applicable to black hole and selective forwarding attacks is presented in [3], where every sensor node sends acknowledgements back to the sink, when a packet is received. That way, the sink can identify locations, where packets get lost and make preparations to avoid these black holes.

2.3 Reacting on misbehavior

While the intrusion detection system in [2] makes no effort do deal with intrusion other than raise a warning, other works show methods of making the network robust and more tolerant to misbehaving nodes. The system for "location-centric isolation of misbehavior" in [3] uses a blacklist embedding in packet headers to route package around areas with misbehaving nodes. The "statistical en-route filtering" in [1] attempts to drop measurements that have not been verified by a number of nodes on the route to the sink.

Chapter 3

An intrusion detection system for wireless sensor networks

This chapter summarizes the architecture and results presented in [2] as an example for a method to gather and analyze information for intrusion detection in wireless sensor networks.

Section 3.1 displays some common information about intrusion detection. Section 3.2 presents the general architecture of the IDS for sensor networks. Section 3.3 describes the rule setup that was used to detect anomalies. Finally, the sections 3.4 and 3.5 show the setup and the results of the evaluation.

3.1 Ordinary intrusion detection

Traditional intrusion detection systems (IDS) can be either net-based or host-based. A net-based IDS monitors the network traffic for unusual behavior while a host-based IDS analyzes the files and programs on the computer itself in order to find out if it has been infiltrated.

Another way to define intrustion detection systems is by their data analysis approach. A behavior-based IDS tries to learn the *normal* traffic and then detects deviations of this normal behavior. A pattern-based IDS recognizes specific attack patterns. The advantages and disadvantages are discussed in more detail in [36]

3.2 Architecture

In sensor networks, since a potential attacker has physical access to the sensor nodes, a host-based intrusion detection system would only be able to recognize the intrusion when the attacker is already capable of reprogramming the whole node or extracting the secret keys. That is why the presented IDS for sensor networks is implemented

net-based in the sense that some of the sensor nodes use promiscous listening to monitor and analyse the network traffic between the nearby sensors.

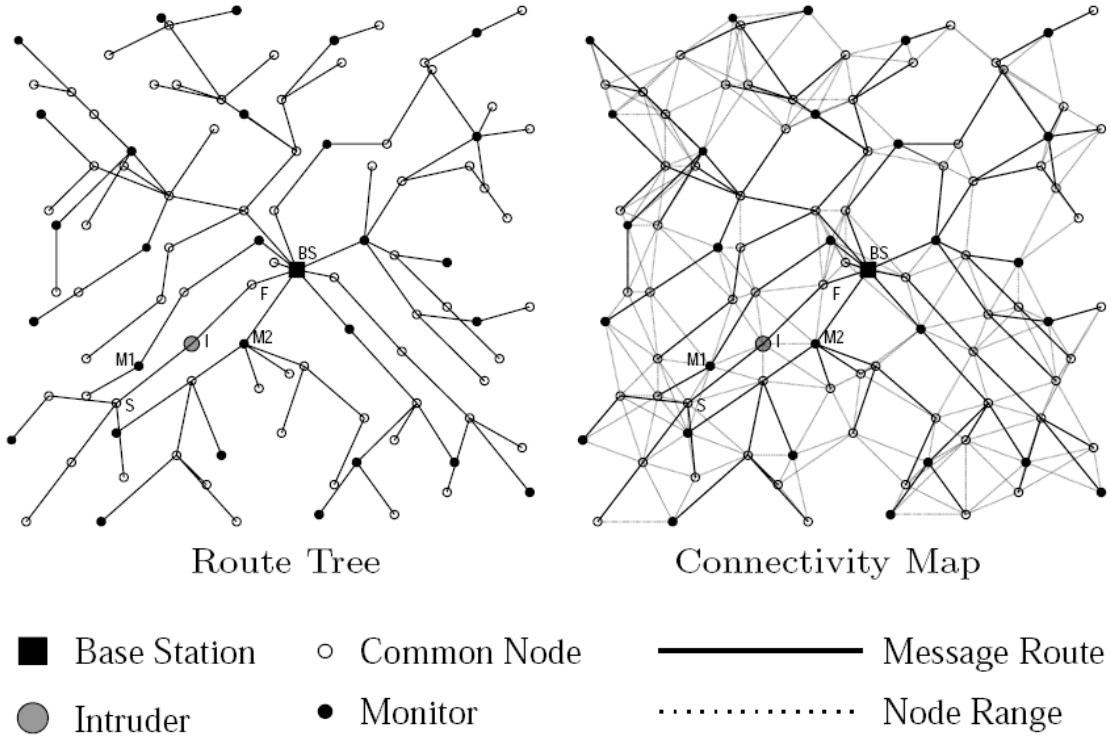

Figure 3.1: Route tree and connectivity map of an example IDS setup ([2])

The system described is pattern-based in the sense that a static set of rules is used to detect malicous behavior. Since attacks have similar indicators as occasional network failures (see section 2.1), the system only raises an alarm if the number of rule applications exceeds a treshold. This treshold is determined during a learning phase directly after deploying the sensors. Thus partly, the system also uses the behavioral paradigm.

It is important to mention that the IDS assumes that

- once deployed, nodes don't change their position anymore and that
- every node is uniquely identifyable.

3.3 Detection algorithm

While listening in promiscous mode, the monitor nodes store important messages in a message buffer until a rule can be applied. From the attacks defined in 2.1, the IDS using the following rules to detect intrusion.

- **Interval rule:** Limits are defined for the minimal and maximal allowed time between two consecutive message of a sending node. For a time below these limits, an exhaustion attack is assumed. [1]
- **Retransmission rule:** The monitor checks whether the nodes in its listening range forward received messages. If a node receives a message, but does not forward it, a blackhole or a selective forwarding attack is assumed.
- **Integrity rule:** The monitor checks for each node whether the payload of the received message matches the payload of the forwarded message. Otherwise a data alteration attack is assumed.
- **Delay rule:** An upper limit is defined for the time in which a node must forward a received message. Otherwise a message delay attack is assumed.
- **Repetition rule:** An upper limit is defined for the number of times that a node may send the same message.
- **Radio transmission range rule:** A message received by the monitor node must originate from a node within the radio transmission range. Otherwise a wormhole attack is assumed. The nodes within transmission range can be identified during the learning phase.
- **Jamming rule:** The number of message collisions must remain under a certain treshold. Otherwise, a jamming attack is assumed.

Each time a rule applies to the events in the network, a counter is raised and if the average number of failures exceeds the treshold determined during the learning phase, an intrusion detection is raised.

3.4 Simulation setup

The system has been tested in a simulator with a setup of 100 sensor node, 28 of which also acted as a monitor node. Figure 3.1 shows the routing tree and the connectivity tree of an example setup. The simulation was divided in 10000 iteration. It began with a 1000 iteration learning-period and continued with the intruder being in turn idle for 700 iterations and then attacking for 200 cycles. Network failures [2] were simulated with a probability of 10% (20% in another simulation). In each

[1] At this point, [2] also assumes a message negligience attack for a time above these limits. This attack is never mentioned again and thus omitted here.

[2] Data alteration, message loss, message collision

simulation, the compromised node attacked with one specific attack[3]. The size of the message buffer was set to 30, 60, 100, 200 and 400 messages.

For each attack, the number of false positives and the percentage of detected attacks (detection rate) was measured.

3.5 Simulation results

This section will briefly display the most important simulation result presented. A complete analysis can be found in [2].

3.5.1 The size of the message buffer

An important result is the relation between message buffer size and the efficiency of the IDS. The message buffer size is a measure of the resource overhead needed, so this relation actually represents the trade-off between resources and efficiency.

- **False positives:** A result of the simulations was, that the number of false positives greatly depends on the size of the message buffer. Smaller buffer sizes create more false positives. The given explanation was, that the influence of the simulated network failures is higher with small buffer size, because the variance is bigger over a small set of samples.

- **Delay attack:** The recognition rate of the delay attack was almost proportional to the buffer size. This seems logical, since messages have to be saved for a long time in order to recognized this attack.

- **Data alteration:** The data alteration attack had such a high number of false positives for small buffer sizes, that a detection is hardly significant. The detection rate was higher for smaller buffer sizes. The explanation in the paper is: "This happens because on larger buffers the generated failures of an attacker do not take the averages of occasional network failures too high" [2, page 21]. There is a statement[4] about this topic in section 5.1.

3.5.2 Reliably detectable attacks

Independently of the size of the message buffer, the detection rate of the **message repetation**, **wormhole** and **blackhole** attack were consistently over 90%. The

[3]Message delay, message repetition, wormhole, jamming, data alteration, blackhole or selective forwarding

[4]I admit that I do not understand the explanation, which is why I provide some personal remarks.

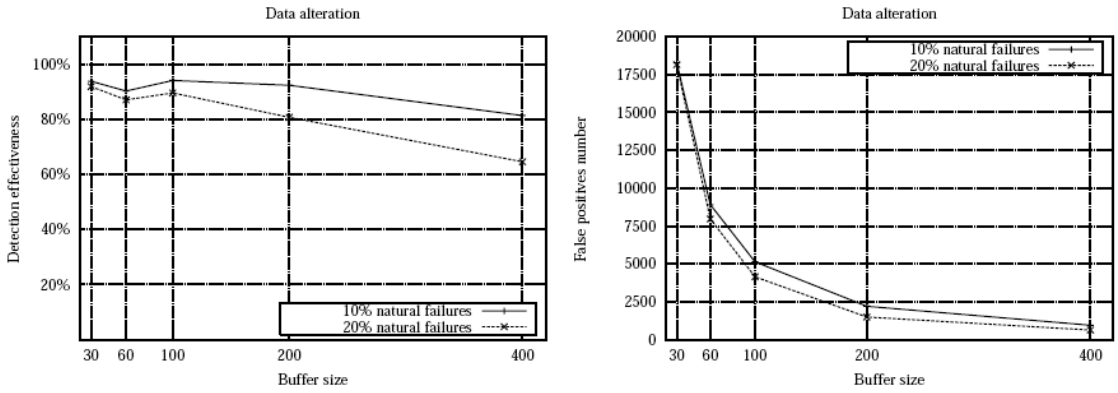

Figure 3.2: Detection effectiveness and false positives for selective forwarding ([2])

detection of the **selective forwarding** attack depended only a little on the buffer size, with a detection rate of over 80% for buffer sizes bigger that 60. Figure 3.2 shows this relation.

3.5.3 Falsely identified attacks

There were some cases, where attacks where mistakenly reported as a different attack. For small buffer sizes, the delay attack was frequently reported as a blackhole attack. The jamming attack was in some cases, mistakenly identified as a blackhole attack, because another node was unable to send any data due to the jamming.

The ability of a monitor to identify an attack depends on the position of the monitor and on the position of the compromised node. It is suggested to distribute the monitors so that for every edge in the routing tree at least one monitor can observe both the sender and the receiver.

3.5.4 Energy consumption

The energy consumption of sensor and monitor nodes was simulated, but not described in greater detail in the paper. It was approximated for the **transmission** and **reception** of message and for **listening** on the network. While "reception" means the full processing of a message, "listening" includes only the verification of the receiver address in the message header of each message. The energy values for each actions were computed for 36 byte message as $Q_{transmission} = 0.48375\frac{mJ}{message}$, $Q_{reception} = 0.1575\frac{mJ}{message}$ and $Q_{listening} = 0.00875\frac{mJ}{message}$.

The main result was, that monitor nodes consume more energy then standard sensor nodes and that the energy consumption is generally higher near the sink.

Chapter 4

Statistical en-route filtering of injected false data

This chapter presents the "statistical en-route filtering of injected false data" described in [1].

The goal of en-route filtering is to identify fabricated reports created by compromised nodes. This identification should at least happen at the sink, but of course it is more energy-efficient if an intermediate node is already able to drop a false report.

The general idea of this work is to distribute the nodes in the network in a density, that a number sensors can always confirm the readings made by another sensor.

Only synchronous cryptography is used, but a special key-distribution scheme allows nodes to falsify reports. This scheme is described in section 4.1. The algorithms to generate reports and the identification of false reports are presented in 4.2 and 4.3 respectively. Finally, section 4.4 presents the methods and results of the evaluation.

4.1 The key distribution scheme

A key distribution scheme has been introduced along with the en-route filtering system. The goal of this scheme is to give every node enough information to identify a false report with a certain probability, yet not enough information to let an attacker create false report by compromising one or two nodes.

The approach is to generate a set of n keys for Message Authentication Codes. The set is then partitioned into m non-overlapping subsets (categories) of size $\frac{n}{m}$. A valid report has to be signed with exactly one key from a predefined number $T \leq m$ of categories. Every key k is assigned an index number i_k which is unique in its category.

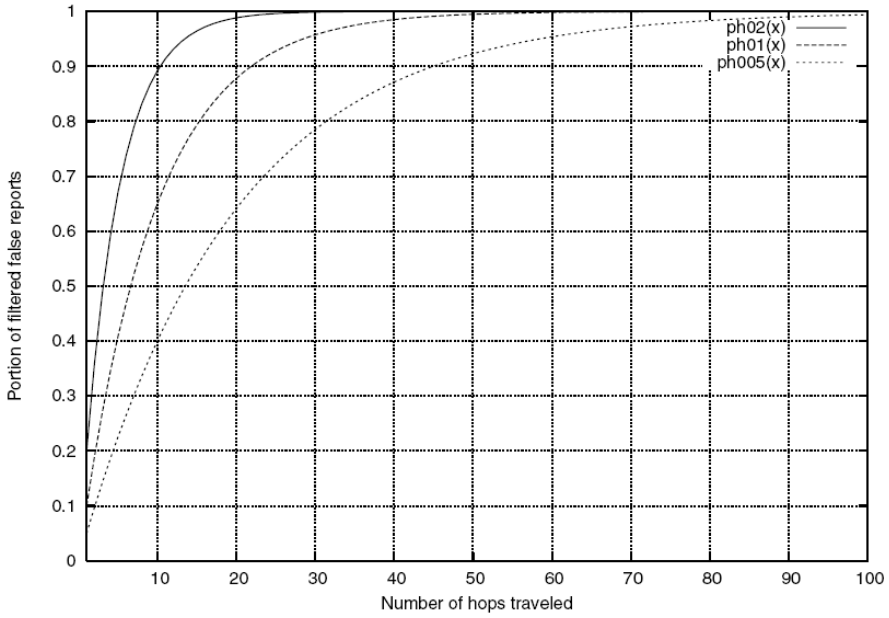

Figure 4.1: Number of identified false report after n hops, for 1,2 and 5 compromised categories [1]

Before deploying the network, a number randomly chosen keys **from the same partition** is stored in each sensor node. There is a remark on that in section 5.2.1.

4.2 Report generation

When a stimulus is detected by some nodes, the node with the strongest reading is elected as the center of stimulus (CoS). It creates the report and sends it to the neighboring nodes, which confirm the reading. Each node return the index i of a random key in their storage key and a Message Authentication Code (MAC) M_i back to the CoS. M_i is computed from the report and the key with the index i. The CoS then attaches one MAC out of each category to the report and sends it towards the sink. The final report has the structure $(R, i_1, M_1, i_2, M_2, \ldots, i_T, M_T)$

In [1], bloom filters are used to reduce the overhead caused by the attached MACs by merging multiple MACs into one value. A method is presented there but ommited here due to manners of space.

4.3 En-route filtering

Every forwarding node checks, whether T keys are attached to the report and whether they are all from distinct categories. If not, the report is dropped. Further-

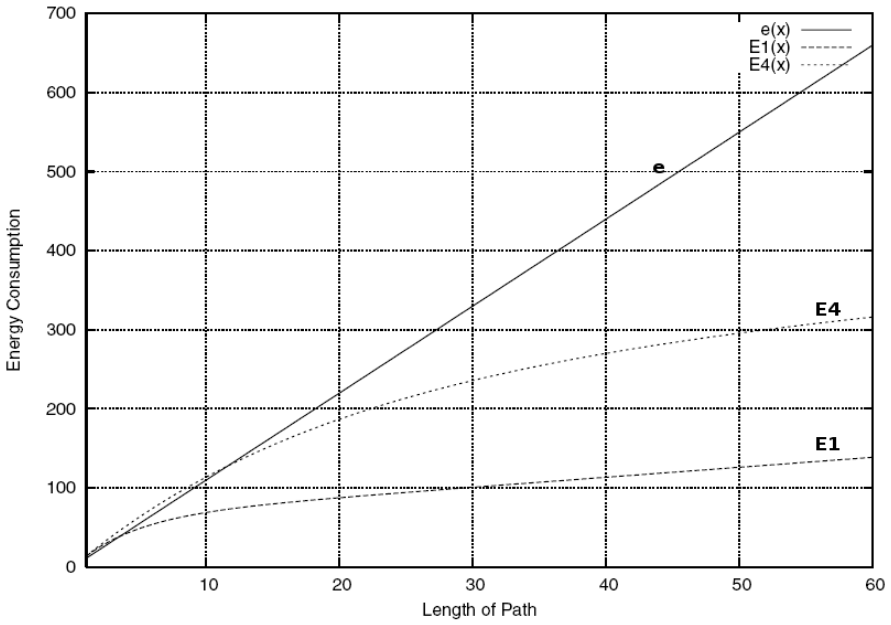

Figure 4.2: Energy consumption without en-route filtering (e), with one compromised category($E1$) and with for compromised categories ($E4$) [1]

more, if one of the attached index numbers is stored in the node, it only forwards the report after verification of the corresponding MAC. If no index number is found, the report is forwarded without check.

In the end, the sink verifies all MACs attached to the report.

4.4 Evaluation

The en-route filtering was simulated in a scenario of 340 nodes, 1000 keys in 10 partitions and 50 keys per node. Figure 4.2 shows the efficiency of the system. With one category compromised by the attacker, 90% of the false reports were dropped within 10 hops.

The energy consumption was also simulated based on the assumption that the transmission of a packet cost 60mW and the reception cost 12mW. Figure 4.4 displays the energy savings due to the en-route filtering.

Chapter 5

Assessment and further ideas

In this chapter I will present my own comments and remarks about both described systems. Additionally I will outline ideas for enhancing and integrating both systems.

5.1 Improvements to the IDS

This section describes some possible improvemens to the intrusion detection system presented in chapter 3.

Detecting data alteration The detection of data alteration attacks was rather unsuccessful in the IDS due to the high number of false positives. Unfortunately, the authors did not state, whether the simulated nodes were using any error-correction schemes in their link layer protocol. However, an error-correction based on CRC and resend-requests still leaves the connection from the supervised node to the supervising monitor open to transmission errors. A monitor cannot tell the node to resend a packet, because it is not the actual receiver. It is not explicitly said, whether the monitor performed a CRC check prior to verifying the forwarded message, but the high number of false positives indicates that it was not done.

I suggest that the monitor discard messages that do not pass a CRC check. An additionally applied forward-error-correction scheme could enable the monitor to correct at least a percentage the messages before checking the payload for data alteration.

Infiltrating the IDS The attacker can as easily compromise a monitor node as a normal sensor node. This possibility is completely ignored in the paper. Using the identity of a monitor, an attacker could raise false alarms as well as prevent alarm

messages from reaching the sink. This could be partly avoided, if the monitor nodes use a system as described in chapter 4.

Instant infiltration The paper ([2]) mentions the assumption that the attacker may needs some time to infiltrate a node and extract the cryptographic information needed for an attack. This constraint is however never mentioned again. Assuming that the node cannot send any data, while being compromised, that should be detected by the monitor nodes as well, maybe as blackhole attack.

Dedicated monitor nodes It may be a good idea to deploy dedicated monitor nodes among the sensors, which are not involved in the routing of normal messages. Since these monitors would not send any data unless an intrusion takes place, they would be harder to find by a potential attacker.

Furthermore, the energy consumption of a monitor node might even be less then consumption of a normal node, if it does not participate in the routing. The monitor would only send data, if an intrusion is detected and according to section 3.5.4, sending consumes three times the energy of receiving. Still, this would have to be tested, because monitors have to *receive* all message instead of just *listening* to them.

5.2 Remarks on the en-route filtering

This section contains a remark on the key distribution scheme presented in section 4.1, as well as a possible enhancement of the en-route filtering system to increase the tolerance against blackhole and selective forwarding attacks.

5.2.1 Key distribution

The key distribution presented in chapter 4 suggests that not all the keys of a partition be stored in a node. The reason is, that fewer keys are compromised if the node is compromised. From the formulars and the description of the system however, it does not matter how many keys of each category are compromised in order to create a false report. One key of each category is enough to make the report seem valid. If no key revocation system is implemented, which is not the case in [1], a category size of 1 which 1 key per node would be the most efficient use of resources with no reduction of detection efficiency.

5.2.2 Countering blackhole and selective forwarding

The en-route filtering system does not address the problem of blackhole and selective forwarding attacks, but they might not be so hard to counter. The en-route filtering system makes the assumption that the density of sensors in the network is high enough to enable many sensors to confirm a generated report. Assuming, the density is also high enough to have more than one possible next hops for each routing decision, it might be possible to implement routing in a fashion that no exact node is adressed, but only a directive like: "My distance to the sink is x, if yours is less, you can forward the message." Then the first receiver would forward the message and all the others could use promiscous listening to find out, that it has already been forwarded correctly. This would be a way to increase to tolerance of the en-route filtering system to blackhole and selective forwarding attacks. On the other hand, the promiscous listening mode would again consume more energy, so this would have to be tested an evaluated as well.

5.3 Combining en-route filtering with intrusion detection

The two systems presented in chapter 3 and 4 complement one another in the sense that the only attack not detected by the intrusion detection system, injection of false data, is handled by the en-route filtering system. It seems reasonable to combine both ideas in order to create a system that is more tolerant to intrusions.

This should be feasible. The only problem is that the IDS could not implement rules to detect black hole and selective forwarding attacks due to the following reasons:

- The **message loss rule** wopuld occur every time a message is dropped because of an invalid MAC
- The monitor node might not be able to distinguish a correctly dropped message from a blackhole attack, because it does not neccessarily have all the keys to verify the MACs of the drop message.
- If every monitor would store all existing MACs, the MACs could be verified, but compromising a single monitor node would be sufficient for an attacker to create false reports.

These attacks would have to be dealt with another way, for example using the scheme suggested in section 5.2.2.

5.4 Unaddressed problems

None of the papers mentioned in this work handles the detection of wrong aggregation behavior. It seems to be a bit more complicated and energy consuming than the presented attacks, but it is an important topic in my eyes.

Another interesting topic in would be a method to hide intrusion alarms from the attacker. If an attacker has physically compromised a node then he is in the area and can probably find out that an intrusion alarm has been sent to the sink. There should be a way to raise a silent alarm that only the sink can detect.

Bibliography

[1] Statistical En-route Filtering of Injected False Data in Sensor Networks, Fan Ye, Haiyun Luo, Songwu Lu, Lixia Zhang, UCLA Computer Science Departement, Los Angeles

[2] Decentralized Intrusion Detection in Wireless Sensor Networks, Ana Paula R. da Silva, Marcelo H. T. Martins, Bruno P. S. Rocha, Antonio A. F. Loureiro, Linnyer B. Ruiz, Hao Chi Wong, October 2005, Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks Q2SWinet '05

[3] Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks, Sapon Tanachaiwiwat, Pinalkumar Dave, Rohan Bhindwale, Ahmed Helmy, University of Southern California, Los Angeles

[36] Claudia Eckert. IT Sicherheit - Konzepte, Verfahren, Protokolle. R.Oldenbourg Verlag, 3 edition, October 2004